Module 14: Gestion des demandes dans les applications Web ASP.NET MVC 5

# **Contenu:**

|  |  |
| --- | --- |
|  | [Aperçu du module](https://www.skillpipe.com/api/2.1/content/urn:uuid:4ed7876c-8864-4cf5-91ee-491726dfe322@2020-12-11T17:05:09Z/OPS/html/20486C14.html#P1) |
| **Leçon 1:** | [Utilisation de modules HTTP et de gestionnaires HTTP](https://www.skillpipe.com/api/2.1/content/urn:uuid:4ed7876c-8864-4cf5-91ee-491726dfe322@2020-12-11T17:05:09Z/OPS/html/20486C14.html#P2) |
| **Leçon 2:** | [Utilisation de Web Sockets](https://www.skillpipe.com/api/2.1/content/urn:uuid:4ed7876c-8864-4cf5-91ee-491726dfe322@2020-12-11T17:05:09Z/OPS/html/20486C14.html#P3) |
| **Laboratoire:** | [Gestion des demandes dans les applications Web ASP.NET MVC 5](https://www.skillpipe.com/api/2.1/content/urn:uuid:4ed7876c-8864-4cf5-91ee-491726dfe322@2020-12-11T17:05:09Z/OPS/html/20486C14.html#P4) |
|  | [Examen du module et points à retenir](https://www.skillpipe.com/api/2.1/content/urn:uuid:4ed7876c-8864-4cf5-91ee-491726dfe322@2020-12-11T17:05:09Z/OPS/html/20486C14.html#P5) |

# **Aperçu du module**

ASP.NET MVC fournit des fonctionnalités qui aident à développer des applications Web. Cependant, ASP.NET MVC n'inclut pas les fonctionnalités qui aident à modifier le codage de la sortie. Dans de tels cas, vous devez savoir comment utiliser un module HTTP ou un gestionnaire HTTP, pour faciliter ces requêtes spécifiques. La plupart des applications Web nécessitent une communication bidirectionnelle entre les systèmes client et serveur. Le protocole Web Sockets facilite les communications bidirectionnelles de manière robuste.

### **Objectifs**

Après avoir terminé ce module, vous serez en mesure de:

|  |  |
| --- | --- |
| • | Utilisez des modules HTTP et des gestionnaires HTTP. |
| • | Utilisez des sockets Web. |

# Leçon 1:**Utilisation de modules HTTP et de gestionnaires HTTP**

Les applications exécutent des actions sur une requête dans le pipeline HTTP, avant de rendre une page Web. Vous devez savoir comment utiliser un module HTTP pour implémenter un mécanisme d'authentification personnalisé pour une page Web, avant de rendre la page Web. Vous pouvez également utiliser un module HTTP pour créer du code qui restitue le contenu dans un format de codage autre que celui par défaut qui répond aux besoins de l'application. Parfois, vous pouvez avoir besoin de l'application pour gérer les demandes à l'aide d'une logique d'application qui diffère de la logique de rendu de page ASP.NET intégrée. Dans de tels cas, vous devez savoir comment utiliser le gestionnaire HTTP pour traiter ces requêtes spécifiques.

## **Objectifs de la leçon**

Après avoir terminé cette leçon, vous serez en mesure de:

|  |  |
| --- | --- |
| • | Décrivez comment les modules HTTP interceptent les requêtes Web. |
| • | Créez des modules HTTP. |
| • | Décrivez comment les gestionnaires HTTP interceptent les requêtes Web. |
| • | Créez des gestionnaires HTTP. |
| • | Déterminez quand utiliser les modules HTTP et les gestionnaires HTTP. |

## **Qu'est-ce qu'un module HTTP?**

Un module HTTP est un programme qui exécute la logique d'application sur une page Web, avant qu'ASP.NET ne rende la page Web. Pour comprendre la pertinence d'un module HTTP, vous devez connaître le pipeline de requêtes HTTP. Le pipeline de requêtes HTTP IIS est une séquence de programmes qui s'exécutent sur des requêtes, pour aider l'application à effectuer des tâches. Lorsque l'application reçoit une demande, elle la transmet à la bibliothèque ISAPI isapi\_aspnet.dll et démarre le pipeline d'exécution ASP.NET pour traiter la demande.

Ensuite, la demande passe par l'instance HttpApplication de l'application et la bibliothèque HttpModule.

Les modules HTTP fournissent les fonctionnalités suivantes à une application:

|  |  |
| --- | --- |
| • | Sécurité. Assurer la prise en charge de la logique d'authentification et d'autorisation personnalisée de votre application |
| • | Journalisation et statistiques. Recueillir des informations de bas niveau sur le cycle d'exécution de l'application Web, à des fins de surveillance et de journalisation |
| • | En-têtes et pieds de page personnalisés. Autoriser l'insertion d'informations d'en-tête personnalisées dans la réponse à chaque demande |

Voici une liste des modules HTTP dans ASP.NET:

|  |  |
| --- | --- |
| • | OutputCache |
| • | Session |
| • | WindowsAuthentification |
| • | FormulairesAuthentification |
| • | PasseportAuthentification |
| • | UrlAuthorization |
| • | FileAuthorization |
| • | DefaultAuthentication |

Si vous configurez des modules HTTP dans une application, les modules HTTP s'appliquent à toutes les requêtes HTTP que l'application reçoit. Vous ne pouvez pas configurer un module HTTP pour une page spécifique.

**Question**: Quels sont les avantages de l'utilisation des modules HTTP dans les applications Web?

## **Création de modules HTTP**

Vous pouvez créer un module HTTP en ajoutant la classe CustomModule à votre projet; cette classe implémente l'interface IHttpModule dans votre application.

**Codage d'un module personnalisé**

classe publique CustomModule: IHttpModule

{

public CustomModule ()

{

}

public String ModuleName

{

get {return "CustomModule"; }

}

public void Init (application HttpApplication)

{

context.Response.Write ("<h1> Démo <h1>");

}

}

Dans la classe CustomModule, vous devez implémenter les deux méthodes suivantes:

|  |  |
| --- | --- |
| • | Propriété ModuleName. Fournit un nom d'affichage qui permet à un autre code d'application d'identifier le module HTTP. |
| • | Fonction init. Fournit l'emplacement pour implémenter toute la logique d'un module HTTP. La classe HttpApplication déclenche cette fonction lorsque l'application reçoit une demande. |

**Enregistrement d'un module HTTP pour IIS**

<configuration>

<system.webServer>

<modules>

<add name = "CustomModule" type = "CustomModule" />

</modules>

</system.webServer>

</configuration>

**Question**: Vous avez développé un HttpModule personnalisé. Que devez-vous faire pour vous assurer que votre application ASP.NET utilise le module personnalisé développé?

## **Qu'est-ce qu'un gestionnaire HTTP?**

Les modules HTTP aident à appliquer la logique à toutes les demandes HTTP qu'une application reçoit. La fonction HttpHandler est un moteur de traitement qui traite des requêtes HTTP spécifiques. Par exemple, vous pouvez configurer une fonction HttpHandler pour gérer les fichiers \* .ashx. Vous pouvez mapper une fonction HttpHandler à une URL basée sur l'extension de fichier de la page côté serveur demandée, telle que \* .aspx. La liste suivante décrit certaines fonctions HttpHandler courantes:

|  |  |
| --- | --- |
| • | Gestionnaire de pages ASP.NET (\* .aspx). Il s'agit du gestionnaire HTTP par défaut pour les pages ASP.NET. |
| • | Gestionnaire de service Web (\* .asmx). Il s'agit du gestionnaire HTTP par défaut pour les pages de service Web. |
| • | Générique wgestionnaire eb (\* .ashx). Il s'agit du gestionnaire HTTP par défaut pour tous les gestionnaires Web qui n'ont pas d'interface utilisateur et qui incluent la directive @WebHandler. |
| • | Gestionnaire de trace (trace.axd). Il s'agit d'un gestionnaire qui permet d'afficher les informations de trace de page. |

Vous pouvez utiliser l'extension de fichier \* .ashx avec le gestionnaire Http car:

|  |  |
| --- | --- |
| • | L'extension n'inclut aucune logique de rendu de page. |
| • | L'extension permet aux développeurs d'écrire une logique pour envoyer des réponses aux systèmes clients. |

Vous pouvez utiliser un gestionnaire HTTP personnalisé avec l'extension \* .rss, pour générer du contenu de fil RSS (Real Simple Syndication) pour les demandes des utilisateurs. Vous pouvez également utiliser un gestionnaire personnalisé pour demander des images à partir de la base de données et envoyer les images aux systèmes clients.

**Question**: Quelle est l'utilisation principale du gestionnaire générique HTTP (\* .ashx)?

## **Discussion: Scénarios pour les modules et gestionnaires HTTP**

Considérez les scénarios suivants. Dans chaque cas, discutez avec le reste de la classe pour déterminer si un HttpModule ou HttpHandler convient le mieux au cas.

|  |  |
| --- | --- |
| • | Vous créez une application de partage de photos et vous souhaitez permettre à chaque utilisateur de discuter de photos et de voyages à vélo avec ses amis. Vous devez vous assurer que l'application restitue les photos aux utilisateurs, à partir de la table de la base de données, sans exiger des utilisateurs qu'ils enregistrent d'abord le fichier sur leur système. |
| • | Vous créez une interface de programmation d'application métier (API) basée sur le transfert d'état de représentation (REST) ​​qui nécessite un en-tête HTTP personnalisé, avant d'accepter une demande. |
| • | Vous créez une API d'entreprise pour fournir du contenu au format XML personnalisé. |
| • | Vous développez une application qui nécessite l'enregistrement des informations de diagnostic sur l'en-tête d'une requête HTTP. |

# Leçon 2:**Utilisation de Web Sockets**

Les technologies AJAX évoluent constamment. Le protocole HTTP ne répond pas aux exigences cruciales telles que les mises à jour des informations en temps réel à partir du serveur. Vous devez savoir comment utiliser les sockets Web pour créer une communication bidirectionnelle entre les systèmes client et serveur. Vous devez également savoir quand utiliser le modèle HTTP traditionnel et le protocole Web Sockets, en fonction des besoins de votre application.

## **Objectifs de la leçon**

Après avoir terminé cette leçon, vous serez en mesure de:

|  |  |
| --- | --- |
| • | Décrivez le protocole WebSocket. |
| • | Décrivez SignalR. |
| • | Utilisez SignalR dans une application Web ASP.NET MVC 5. |

## **Qu'est-ce que le protocole WebSocket?**

Le protocole WebSocket est une norme du World Wide Web Consortium (W3C). La spécification HTML 5 inclut la prise en charge des sockets Web.

Le protocole facilite la communication bidirectionnelle entre les navigateurs Web et les serveurs Web. Le protocole permet de surmonter le problème de HTTP ne fournissant pas de communication bidirectionnelle.

Les développeurs créent souvent des applications de communication bidirectionnelle en utilisant des techniques inefficaces telles que de longues boucles et des interrogations. Ces techniques consomment de grandes quantités de mémoire et d'autres ressources des systèmes serveur et client. Le protocole WebSocket résout les problèmes liés à l'utilisation de ces techniques en créant une connexion socket constante entre le client et le serveur. Toutes les communications entre les systèmes client et serveur se font alors via le socket. Ce socket élimine le besoin de recréer une demande plusieurs fois, pendant les communications en temps réel entre les systèmes client et serveur. Les navigateurs plus anciens peuvent ne pas prendre en charge les sockets Web.

Les sockets Web fonctionnent de la même manière que les sockets réseau traditionnels. La seule différence est que lors de la négociation initiale, les sockets Web utilisent la requête HTTP Upgrade qui inclut l'en-tête HTTP Upgrade. Si le serveur accepte les sockets Web, le serveur renvoie une réponse avec le code d'état 101. Ensuite, les systèmes client et serveur envoient la charge utile à l'aide du socket. Lorsque le système client n'a plus besoin de communiquer avec le serveur, l'application envoie une charge utile de fermeture de connexion, pour avertir le serveur de fermer le socket Web.

**Question**: Quelle est la principale différence entre les sockets HTTP traditionnels et Web?

## **Qu'est-ce que SignalR?**

SignalR est un ensemble de composants qui simplifie le développement d'applications Web bidirectionnelles en temps réel, telles que les salles de discussion sur les sites Web. SignalR utilise les WebSockets dans la mesure du possible pour se connecter. Cela signifie qu'il prend en charge tous les navigateurs compatibles HTML5. Un avantage clé de l'utilisation de SignalR est qu'il prend en charge les appels de procédure à distance (RPC) avec des applications HTML ou même .NET et peut automatiquement basculer vers une interrogation périodique pour les navigateurs plus anciens.

Pour installer le package dans votre application, vous pouvez utiliser NuGet pour rechercher le package, Microsoft.AspNet.SignalR.

Après l'installation, vous pouvez implémenter le code côté serveur pour la logique serveur.

**Utilisation de SignalR**

en utilisant Microsoft.AspNet.SignalR;

ChatHub de classe publique: Hub {

public void Send (nom de chaîne, message de chaîne)

{

Clients.All.broadcastMessage (nom, message);

}

}

**Enregistrement du middleware SignalR**

Démarrage de classe publique

{

Configuration publique void (application IAppBuilder)

{

app.MapSignalR ();

}

}

SignalR comprend également une bibliothèque JavaScript qui peut être utilisée par les clients pour faciliter les communications entre le client et le serveur. La bibliothèque SignalR dépend de la bibliothèque jQuery.

**Code SignalR côté client**

<script src = "@ Url.Content (" ~ / Scripts / jquery.signalR-2.2.2.min.js ")"> </script> <script src = "@ Url.Content (" ~ / signalr / hubs ")"> </script>

<script type = "text / javascript">

$ (fonction () {

var chat = $ .connection.chatHub;

chat.client.broadcastMessage = function (nom, message) {

alerte (message);

});

$ .connection.hub.start (). done (function () {

$ ("# diffusion"). cliquez sur (function () {

chat.send (nom d'affichage, 'Bonjour');

});

});

});

</script>

**Question:**Quel est le principal avantage de l'utilisation de SignalR, au lieu de WebSockets directement?

## **Démonstration: Comment ajouter une salle de conversation à une application Web à l'aide de SignalR**

Dans cette démonstration, vous verrez comment ajouter une salle de discussion simple à une application Web à l'aide de SignalR.

### **Étapes de démonstration**

|  |  |
| --- | --- |
| • | Vous trouverez les étapes dans la section «Leçon 2: Utilisation de Web Sockets» sur la page suivante:<https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD14_DEMO.md>. |

# **Atelier: Gestion des requêtes dans les applications Web ASP.NET MVC 5**

### **Scénario**

Le conseil d'administration et les responsables d'Adventures Works sont satisfaits de l'application de partage de photos, mais ont demandé que l'interactivité soit maximisée pour encourager les utilisateurs à s'inscrire et à participer pleinement à la communauté. Par conséquent, il vous a été demandé d'ajouter une fonctionnalité de chat à l'application. Les membres authentifiés doivent pouvoir démarrer une discussion sur une photo particulière à partir de la vue Affichage. Les salles de discussion pour chaque photo doivent être séparées les unes des autres. Les utilisateurs de la salle de discussion devraient pouvoir envoyer un message à tous les autres utilisateurs de cette salle de discussion et ils devraient pouvoir voir tous les messages qui ont été envoyés depuis qu'ils ont rejoint la salle de discussion.

Vous avez décidé d'utiliser SignalR pour implémenter la salle de discussion sur Web Sockets.

### **Objectifs**

Après avoir terminé cet atelier, vous serez en mesure de:

|  |  |
| --- | --- |
| • | Installez SignalR dans une application Web ASP.NET MVC 5. |
| • | Configurez SignalR sur le serveur et créez un concentrateur SignalR. |
| • | Lien vers les fichiers de script requis pour SignalR dans une vue MVC. |
| • | Créez le script pour les connexions SignalR et envoyez des messages aux groupes. |

##### **Configuration du laboratoire**

Durée estimée: 60 minutes

Vous trouverez les étapes de haut niveau sur la page suivante:<https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD14_LAB_MANUAL.md>.

Vous trouverez les étapes détaillées sur la page suivante:<https://github.com/MicrosoftLearning/20486-DevelopingASPNETMVCWebApplications/blob/master/Instructions/20486C/20486C_MOD14_LAK.md>.

### **Exercice 1: Création d'un concentrateur SignalR**

##### **Scénario**

Avant de pouvoir écrire du code JScript sur le client pour vous connecter à SignalR, vous devez configurer et coder un concentrateur SignalR sur le serveur Web.

Dans cet exercice, vous allez:

|  |  |
| --- | --- |
| • | Installez SignalR dans l'application de partage de photos. |
| • | Configurer le routage. |
| • | Créez un hub SignalR pour accepter les messages des clients et transférer ces messages à d'autres clients qui discutent de la même photo. |

### **Exercice 2: Création d'une vue de chat photo**

##### **Scénario**

Maintenant que vous avez installé et configuré SignalR et un concentrateur SignalR côté serveur, vous devez utiliser JScript et la bibliothèque SignalR JScript pour envoyer et recevoir des messages côté client.

Dans cet exercice, vous allez:

|  |  |
| --- | --- |
| • | Créez une nouvelle action de contrôleur MVC et une vue Razor pour afficher l'interface utilisateur de discussion pour une photo particulière. |
| • | Liez aux bibliothèques JScript dont SignalR a besoin et écrivez un script côté client pour appeler les méthodes Join () et Send () sur le hub. |
| • | Testez la fonctionnalité de chat. |

### **Question (s) de révision**

**Vérifiez vos connaissances**

**Découverte**

**Dans la fonctionnalité de discussion que vous avez créée, chaque photo de l'application Partage de photos dispose d'une salle de discussion distincte. Comment cette séparation est-elle possible avec un concentrateur SignalR?**

Afficher la solution Réinitialiser

# **Examen du module et points à retenir**

La plupart des applications Web ne se limitent pas au rendu de contenu HTML. Ils nécessitent des fonctionnalités qui prennent en charge une logique personnalisée telle que l'authentification personnalisée. Pour répondre à de tels cas, vous pouvez utiliser des modules HTTP et des gestionnaires HTTP pour exécuter une logique personnalisée avant ou après le rendu d'une page Web. La technologie de socket Web de MVC 5 prend en charge la communication bidirectionnelle entre les systèmes client et serveur. Cette technologie est utile pour les applications Web qui nécessitent des mises à jour constantes entre les systèmes client et serveur.

### **Problèmes et scénarios du monde réel**

Vous créez une application qui obtient les dernières informations sur les prix des produits à partir de la base de données interne. La tarification est constamment mise à jour chaque fois que les utilisateurs professionnels ressentent le besoin de mises à jour. Par conséquent, vous devez vous assurer que l'application met à jour les informations de tarification toutes les cinq minutes. Dans de tels cas, vous pouvez utiliser la technologie de socket Web pour implémenter la mise à jour des prix. Vous pouvez également ajouter du code pour télécharger l'image du produit stockée dans la base de données du produit à l'aide d'un gestionnaire générique (fichier \* .ashx).

### **Question (s) de révision**